REMARKS

Reconsideration of this application is respectfully requested. Claims 1-14 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 6,510,497 by Strongin et al. (hereinafter "Strongin").

Claims 1, 3, 6-9 and 12 have been amended. Claim 4 has been canceled without prejudice. New claims 15-21 have been added.

Claim Rejections -35 USC § 102

The Office Action rejects claims 1-14 under 35 U.S.C. 102(e) as being anticipated by Strongin. Applicant respectfully asserts that independent claim 6 is not anticipated under 35 U.S.C. 102(e) by Strongin. The Examiner states:

> With respect to independent claim 8, a scheduler is disclosed as a memory arbiter in column 17, line 4.
> A switch point is disclosed as discussed supra with respect to claims 1 and 6. A current device state is disclosed as a bus direction and/or a page status in column 18, lines 22-35.
> A count is disclosed as "one or more" in column 18, line 30.
> Logic configured to determine an updated device state using the switch point and count such that when the count crosses a threshold of the switch point, the device state is changed is disclosed in column 18, lines 22-35, where memory accesses are scheduled based on the bus direction and the page status.
> Scheduling the access requests to the device using the updated device state is disclosed in column 18, lines 22-35, which discloses that after the pending requests that are consistent with the device state are issued, the access requests are then issued ("ahead of') that are inconsistent with the previous device state.
> With respect to independent claim 1, . . .
> Scheduling requests to a device using the current state of the device, the count of the number of requests that have already been scheduled using the current state, a switch point (number of pending operations) indicating when to switch state, wherein after the count reaches the switch point and there are incoming requests having an alternate state to the current state (a different bus direction or a different open bank of DRAM) of the device, switching the state of the device to process

> incoming requests is disclosed in column 12, lines 20-35. Also see column 18, lines 22-35, which discusses scheduling a number of "tracked" requests based on the bus direction, or "device state." The "switch point" is when the pending requests consistent with the memory bus direction are issued, and the bus direction reverses, or switches, to allow the scheduled requests that were previously inconsistent with the previous bus direction to now issue.

(Office Action pages 3-5) (emphasis added)

The Examiner and applicant have been debating the meaning of the term switch point as used in the claims for the last few office actions. The Examiner's take is anytime a device switches states then a switch point has been satisfied. The applicant has argued that a switch point is a threshold number/count of requests. Applicant believes the language in the claims clearly conveyed this meaning but has amended the independent claims to solidify this point.

Independent claim 6 states:

> 6. A bus scheduler comprising:
> an input configured to receive at least one incoming request, each request indicating a bus direction;
> a switch point;
> an indicator of a current bus direction;
> a unit to track a count of requests processed using the current bus direction; and
> logic configured to switch the direction of the bus to process incoming requests wherein after the count reaches a threshold value established for the switch point and there are incoming requests having the direction opposite to the current direction of the device bus, switching the direction of the device bus.

(emphasis added)

Strongin does not disclose or even suggest establishing "a threshold value for a switch point." Accordingly, Strongin also cannot disclose "after the count reaches a threshold value established for the switch point and there are incoming requests having the direction opposite to the current direction of the device bus, switching the direction of the device bus." Strongin merely suggests examining which memory page is "open" and if possible identifying one or more requests addressed for that memory page in a queue to be serviced. Strongin then services those one or more requests over other potential pending requests by granting a memory access to those requests. Strongin states:

> It has been discovered that the efficiency of memory controller 400 can be enhanced by empowering memory controller 400 such that memory controller 400 can reorder memory transactions to substantially maximize memory efficiency. This approach can, among other things, increase the page-hit rate, thus improving the memory subsystem performance. In one embodiment, memory controller 400 may reorder transactions such that accesses to currently open pages are completed ahead of transactions that are targeted to pages not currently open.

(Strongin Col. 11, Lns. 1-10) (emphasis added)

> In light of the foregoing, in the event that a memory controller has several memory accesses to be done sequentially, then once a page is open it would make sense (but it is not currently done in the art) from an efficiency standpoint to examine pending as well as current memory accesses in order to determine which of those pending memory accesses will be to memory locations that are within a currently open page (that is, the row of the request is the row from which a memory controller is currently reading within a DRAM). In other words, assuming a page X is open, if there are four memory accesses A, B, C, and D, waiting to be performed, and assuming the first access A is to page Z, the second access B is to page X, the third access C is to page Y, and the fourth access D is to page W, it is preferable from a memory efficiency standpoint that the data access (i.e., access B) appropriate to the page that is open (i.e., page X) be made first.
>
> Current memory controllers do not typically "look ahead" to see if certain pending memory accesses are destined for currently open pages.

(Strongin Col. 4, Lns. 5-21) (emphasis added)

As discussed in the previous office actions, Strongin creates categories of requests based on memory page availability and current bus direction. Strongin then services one entire category prior to servicing another category. Strongin states:

> It has been discovered that significant reductions in main memory latency can be achieved by taking advantage of correlations internal to multiple independent streams of memory accesses. As used herein, the term "correlation" means that different addresses corresponding to different accesses tend to fall within a relatively narrow range. For non-limiting example, when AGP-enabled graphics controller 100 accesses system memory 116, such accessing tends to be highly correlated in that the memory locations accessed tend to be in closely situated addresses. The present invention, among other things, allows correlations present to be taken advantage of in order to reduce memory latency.

(Strongin Col. 8, Lns. 35-46) (emphasis added)

> In one embodiment, the pending memory operations are scheduled for execution in the following hierarchy: first, those non-speculative memory operations directed to open pages and of a type consistent with the direction of any data bus associated with those open pages (e.g., a write operation would be scheduled for execution if a bus were pointed from memory arbiter 482 to an open page in system memory 116 to which the write operation indicates data is to be written) are scheduled for execution (and, optionally, following the selection, thereafter reordered on the basis of the relative priorities of the requests selected for execution). Second scheduled for execution are those pending non-speculative memory operations within requested memory operation buffer 136 which are directed toward open pages, but which are of a type which would require that the bus associated with the open pages be reversed in direction (again such requests can be optionally rearranged following selection depending upon any associated priorities). Third scheduled for execution are those non-speculative operations directed toward pages which are not currently open in system memory, but which are consistent with the direction of a bus associated with the target pages (again, optional reordering may be done). Fourth scheduled are those non-speculative requests directed to closed pages and which are inconsistent with the direction of a bus associated with the target pages (again, optional reordering may be done). Fifth scheduled are those non-speculative operations directed to pages in a refresh state (again, optional reordering may be done). Last scheduled are speculative operations or accesses.

(Strongin Col. 12 Lns 20-39) (emphasis added)

Strongin clearly discloses that the memory arbiter 482 schedules access to the system memory 116 for request on the bus via a category based hierarchy. Strongin creates these categories of requests based at least on memory page availability and current bus direction. All of the requests in a specific category of request are serviced prior to servicing requests in a second category. Strongin does not disclose servicing a threshold number of request in a specific category until a switch point is satisfied and then servicing requests in a second category if the requests in the second category are present. Strongin does not disclose or even suggest establishing "a threshold value for a switch point." Accordingly, Strongin also cannot disclose "after the count reaches a threshold value established for the switch point and there are incoming requests having the direction opposite to the current direction of the device bus, switching the direction of the device bus."

Therefore Strongin does not disclose every limitation in claim 6. As such, independent claim 6 is not anticipated under 35 U.S.C. 102(e) by Strongin.

Given that claims 7, 12, 16, and 20 depend from and include the limitations of claim 6, claims 7, 12, 16, and 20 are also not anticipated under 35 U.S.C. 102(e) by Strongin.

Likewise, independent claims 1, 8, and 21 are not anticipated under 35 U.S.C. 102(e) by Strongin for similar reasons as discussed above.

Given that claims 2, 3, and 5 depend from and include the limitations of claim 1, claims 2, 3, and 5 are also not anticipated under 35 U.S.C. 102(e) by Strongin.

Given that claims 9-11, 13-15, 17, and 19 depend from and include the limitations of claim 8, claims 9-11, 13-15, 17, and 19 are also not anticipated under 35 U.S.C. 102(e) by Strongin.

The Office Action also states:

With respect to claims 3 and 12, the switch point being adjustable by software is disclosed in column 15, lines 5-10.

However, Strongin actually merely discloses that the functions of the components discussed in Strongin may be implemented in hardware or software. Strongin is silent regarding the switch point being adjustable by software. Strongin states:

> The foregoing detailed description has set forth various embodiments of the present invention via the use of block diagrams, pictographic representations, flowcharts and examples. It will be understood as notorious by those within the art that each component, step, and operation illustrated by the use of block diagrams, pictographic representations, and examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof. In one embodiment, the present invention is implemented via Application Specific Integrated Circuits (ASICs). However, those skilled in the art will recognize that the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard Integrated Circuits, as a computer program running on a computer or processor, as firmware, or as virtually any combination thereof and that designing the circuitry and/or writing the code for the software or firmware would be well within the skill of one of ordinary skill in the art in light of this specification.

(Strongin Col. 14, Ln. 60 to Col. 15, Ln. 11) (emphasis added)

Therefore Strongin does not disclose the limitations in claims 3 and 12. Thus, claims 3 and 12 are not anticipated under 35 U.S.C. 102(e) by Strongin.

Also, Strongin does not disclose or suggest the memory controller 400 may be implemented on in a System on a Chip. Strongin merely discloses that the memory controller 400 may be implemented in a desk top personal computer environment where the DRAM is located on an entirely different card then the processor and other internal computer circuits. (See Strongin Col. 6, Lns. 10-50 and Figure 1 as well as Col. 1, Lns. 49-61).

Therefore Strongin does not disclose the limitations in claims 17 and 18. Thus, claims 17 and 18 are not anticipated under 35 U.S.C. 102(e) by Strongin.

Also, Strongin does not disclose or suggest that request may have Quality of Service guarantees associated with those requests.

Therefore Strongin does not disclose the limitations in claims 19 and 20. Thus, claims 19 and 20 are not anticipated under 35 U.S.C. 102(e) by Strongin.

Conclusion

It is respectfully submitted that in view of the amendments and remarks set forth herein, the rejections and objections have been overcome. Applicants reserve all rights with respect to the application of the doctrine equivalents. If there are any additional charges, please charge them to our Deposit Account No. 02-2666. Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 7-5-05

Thomas S. Ferrill
Reg. No. 42,532
Tel.:(408) 720-8300

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026